PERT/CPM

Quick and Simple Critical Path Solution Heuristic

Military Operations Research Society Symposium

June 2009
Fort Leavenworth, Kansas

by
Michael S. David
Associate Professor of Operations Research
U.S. Army Logistics Management College
School of Systems and Acquisition Management
Management Science Committee
Fort Lee, Virginia

TABLE OF CONTENTS

	Page
Introduction	**1**
• Abstract	1
• Tutorial Topics	1
• Managerial Relevance	1
Rudiments of Forward Pass/Backward Pass	**2**
• Network Symbols and Protocol	2
• Activity Duration	2
• Forward Pass/Backward Pass Algorithm	3
Forward Pass	3
Network Critical Path Duration	5
Backward Pass	5
Network Critical Path Determination	7
The Logic Method	**8**
• Technique Algorithm	8
• Longest Path Logic Application	8
• Determining Alternate Paths Slack	12
Forward Pass/Backward Pass Method	13
Logic Method	14
Advantages and Disadvantages of Both Techniques	**15**
• Forward Pass/Backward Pass Method	15
Advantages	15
Disadvantages	15
• Logic Method	15
Advantages	15
Disadvantages	15
Index	16

PERT/CPM

Quick and Simple Critical Path Solution Heuristic

Introduction

Abstract

The Program Evaluation and Review Technique (PERT) and Critical Path Method (CPM) have been useful project management tools since their development in the 1950s. The ease of use and utility of these tools have made them mainstays in project modeling and management over the years. They can be found readily on the job and in the college classroom when project planning is addressed.

Tutorial Topics

- Rudiments of Forward Pass/Backward Pass
- Logic Method Heuristic
- Solving for the Critical Path with the Logic Method
- Validating the Logic Method
- Logic Method Application to Activity on Arc and Activity on Node Networks

Managerial Relevance

Although the PERT Network technique can be a valuable and simple tool in managing a project, it is frequently shunned for more sophisticated computer driven techniques that appear to be easy and simple to use. This disuse may be grounded in PERT's lack of sophistication, but the difficulty and cumbersome method used to solve and manage a PERT Network is a major reason. This paper covers a new technique, the Logic Method, that is easy to learn and use. This will enable the user to more quickly apply the PERT Network technique to their project and reap its rewards of more prudent and effective project management with minimal effort.

Rudiments of Forward Pass/Backward Pass

Network Symbols and Protocol

> To simplify this presentation, unless otherwise specified all networks discussed will be the Activity on Arc type.

A **PERT/CPM Network** has the following three components:

- **Nodes** – circles that represent activity completion mile-stone events and are uniquely identified with a number 7
- **Activity** – solid lines with arrows, which show ⟶ direction, that represent activities and are identified by their nomenclature or abbreviated names
- **Dummy Activity** – dotted lines with arrows, which show - - - - - - - -▶ direction, that create unique paths or carry messages of activity completion to selected parts of the network

Each Activity consumes a resource (time and/or money); where each **Dummy Activity** consumes no resources since they provide flow logic and messaging within the project network model.

Note: An **Activity** is synonymous to an **Arc** which is the popular description used on most texts.

Activity Duration

Each **Activity** or **Arc** (task) is represented by a line which is known as an Activity. The time it takes to complete each activity is known as its duration. Each **Activity** starts at a node which represents when it begins and ends at a node which represents it completion. Since multiple Activities (i.e. activities) can begin and end anywhere in a project, it is necessary to account for each **Activity's duration** in order to determine when everything begins and ends.

All activities that begin at the network initial, or start, node start at time zero. The duration of each of these Activities is than added to zero to determine when the activity is completed. Start and finish times for **Subsequent Activities** are then based on downstream completion times determined with the **Forward Pass/Backward Pass Algorithm**.

Example PERT/CPM Network

The model shown in Figure 1 on the next page demonstrates the logic of a **PERT/CPM Network**. The logic flow begins at **Node 1** and terminates at **Node 14** with **17 Activities** represented by **arrowed lines** and **8 Dummy Activities** represented by arrowed **dotted lines**. Each **Activity** is identified by a **letter name** displayed above its **Arc. Activity Time** is represented by a **numerical value** shown below the **Arc**. The names can be referenced to a table that defines the description of each activity.

Since all activities that begin at the **Initial Node** (Node 1) start at time zero they end after their duration. Therefore **Activity A** is completed in **0 + 15** or **15 days**; **Activity B** is completed in **0 + 10** or **10 days**; and **Activity C** is completed in **0 + 50** or **50 days**.

The start and finish times for all activities that do not originate from the initial node depend on when the nodes that launch them have all their logic requirements satisfied. For example, **Node 4** is waiting for the **Dummy Activity** that represents the **completion** of **Node 8** which is waiting for the **Dummy Activity** that represents the **completion** of **Node 2**. **Activity D** cannot begin until the milestone event **Node 4** happens.

Monitoring this process is tedious and cumbersome but not complicated. The traditional technique used to determine the start and finish times of all activities and the project is known as the **Forward Pass/Backward Pass Algorithm** which is covered below.

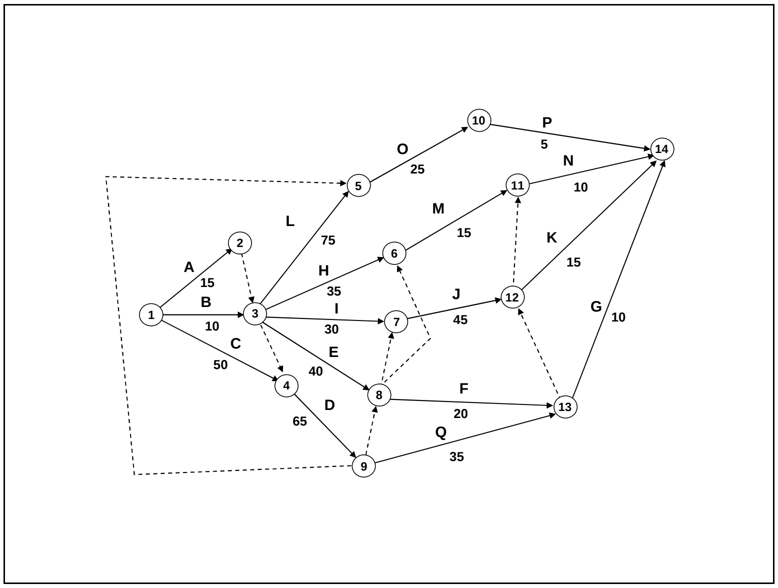

Figure 1
PERT/CPM Example Network

Forward Pass/Backward Pass Algorithm

This concept is based on the following four events for each activity:

- **Early Start** – the **soonest** an activity can **start**
- **Early Finish** – the **soonest** an activity can be **completed**
- **Latest Start** – the **deadline** of when an activity **must start** without affecting the project schedule
- **Latest Finish** - the **deadline** of when an activity **must be completed** without affecting the project schedule

Forward Pass

This part of the algorithm deals with the **Early Start (ES)** and **Early Finish (EF)** times. These values are determined for each activity as follows:

Early Start = **Largest Early Finish** of **All Activities that end** at an **Activity's Launch Node**

Early Finish = **Early Start + Activity Duration**

The **Forward Pass** starts at the **Network Initial Node** (beginning of the project) where the **Early Start** for all activities launched from there is always **zero**.

For the purposes of brevity, the following, which will be used in this presentation, are abbreviated versions of the above formulas:

ES = Largest Input EF

EF = ES + Duration

For tracking purposes, these values are placed **above each Activity** with **ES next** to the **launch node** and **EF next** to the **completion node**.

An example of the **ES** and **EF** for each activity in a typical network is shown in Figure 2 below.

Note that **Activities A**, **B** and **C** start at the beginning of the project and therefore each has en **ES** of **zero**. Since **A's duration** is **15**, its **EF = ES + 15** or **15**. Similarly **B's EF** is **10** and **C's EF** is **50**. Each of these **EF values** is placed on top of its **Activity** next to its completion node.

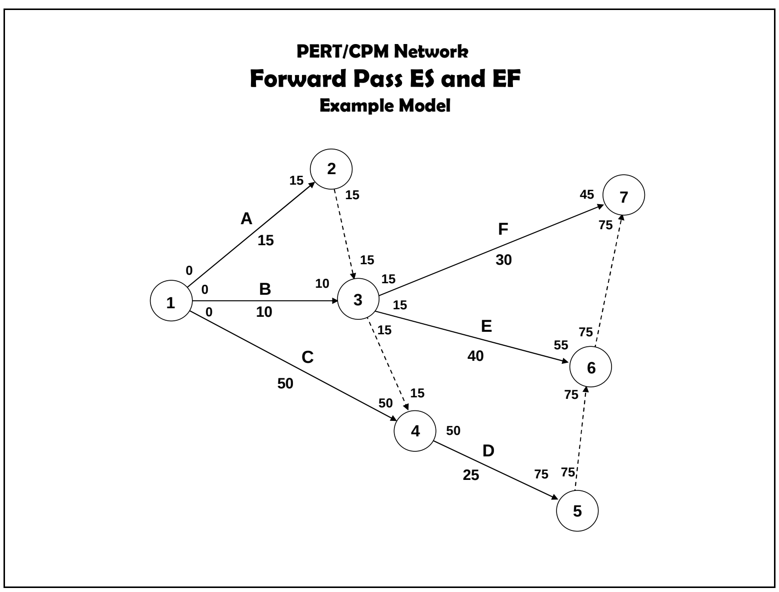

Figure 2
Forward Pass Example Model

Since only **Activity A** is completed at **Node 2**, its **EF** of **15** becomes the **ES** for the **Dummy Activity** launched at **Node 2**. This **Dummy Activity** has **no duration** therefore its **EF = 15 + 0** or **15**.

The **ES** for **all the Activities** launched at **Node 3** is **15** which is the **Largest EF (15** and **10)** of the **two Activities** that finish there (**Dummy Activity** and **Activity B**). This process continues until all **Activities** have the **ES** and **EF** shown in Table 1 below.

Activity	ES	EF
A	0	15
$\text{Dummy}_{N2\text{-}N3}$	15	15
B	0	10
$\text{Dummy}_{N3\text{-}N4}$	15	15
C	0	50
D	50	75
$\text{Dummy}_{N5\text{-}N6}$	75	75
E	15	55
$\text{Dummy}_{N6\text{-}N7}$	75	75
F	15	45

Table 1
Example Model Activities ES and EF

Network Critical Path Duration

Since the **Largest Early Finish** of **any activity** that **ends** at the **Network Terminal Node** represents the **soonest** the project can be completed, its value is the **duration** of the **Critical Path**. This is expressed in the following formula:

Critical Path Duration = Project Earliest Finish

Project Earliest Finish = Largest EF of any Activity ending at the Network Terminal Node

In the **Forward Pass** example above, shown in Figure 2 and Table 1, the **duration** of the **Network Critical path** is **75** since it is the **largest EF** (45 and 75) that finishes at **Node 7** which is the **Terminal Node** for the project.

Backward Pass

This part of the algorithm deals with the **Late Finish (LF)** and **Late Start (LS)** times of each **Activity**. These values are determined for each activity as follows:

Late Finish = **Smallest Late Start** of **All Activities** that are launched at the **Activity's Completion Node**

Late Start = **Late Finish – Activity Duration**

The **Backward Pass** is initiated at the **Network Terminal Node** (end of the project) where the **Late Finish** for all activities ending there is always the **Duration of the Critical Path** which is the **final deadline** for the project.

For the purposes of brevity, the following, which will be used in this presentation, are abbreviated versions of the above formulas:

LF = Smallest Completion Node Output LS

LS = LF - Duration

For tracking purposes, these values are placed **below each Activity** with **LS next** to the **launch node** and **LF next** to the **completion node**.

The **Late Finish** for all activities that end at the **Network Terminal Node** (end) is always the **Smallest Early Finish** of the activities that end there.

Using the same **Forward Pass** example above and shown again in Figure 3 below, the **LS** and **LF** for each activity are determined in the **Backward Pass** as shown below. The value in boxes, $\boxed{75}$, are the **LS** and **LF** times.

Note that **Activity F** and a **Dummy Activity** end **Node 7** which is the **Terminal Node** of the project. Based on the rule dealing with "… the **Largest Early Finish** of **any activity** that **ends** at the **Network Terminal Node**…**"** under **Network Critical Path Duration** on **page 5**, the **duration** of the example **Network Critical Path** is **75**.

This makes the **LF** for **Activity F** and the **Dummy Activity** equal **75**. Each of these **LF values** is placed below its **Activity** next to **Node 7**.
.

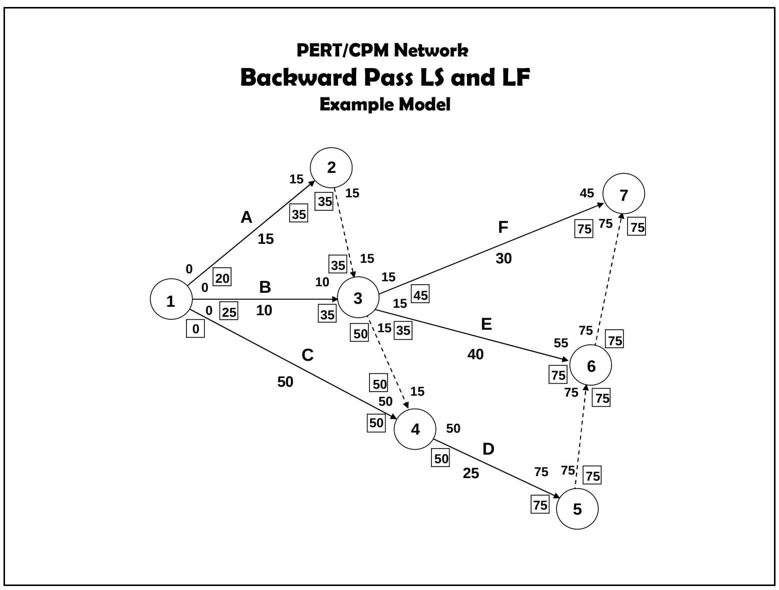

Figure 3
Backward Pass Example Model

Since the duration of **Activity F** is **30**, its **LS = 75 – 30** or 45. This value is placed below **Activity F** next to **Node 3** which is where it begins. The **Dummy Activity** has **no duration** therefore its **LS = 75 - 0** or **75** which is placed below the **Dummy Activity** next to Node 6 where it begins.

This process continues until all **Activities** have the **LS** and **LF** shown in Table 2 on the next page. Note that the **LF** for **all the Activities** that end at **Node 3** is **35** is the **Smallest LS** of the **three Activities (45, 35 and 50)** that begin there (**Dummy Activity, Activity E** and **Activity F**).

Activity	LS	LF
A	20	35
$Dummy_{N2-N3}$	35	35
B	25	35
$Dummy_{N3-N4}$	50	50
C	0	50
D	50	75
$Dummy_{N5-N6}$	75	75
E	35	75
$Dummy_{N6-N7}$	75	75
F	45	75

Table 2
Example Model Activities LS and LF

Network Critical Path Determination

Once **Early Start (ES)**, **Late Start (LS)**, **Early Finish (EF)** and **Late Finish (LF)** have been determined, the **Critical Path** can be ascertained**.** This is done by **comparing** the **ES** and **LS** as well as the **EF** and **LF** for each **Activity**. Each **Activity** with **Start values equal** and the **Finish Values equal** is on the **Critical Path**. These **Activities** will form a **continuous path** between the network **Initial** and **Terminal Nodes**.

If there is a gap in the path, a mistake was made when determining the **Start** and **Finish values.**

In the above example the **Critical Path**, shown in **bold** in Figure 4 below, is the following:

Activity C – Activity D – $Dummy_{N5-N6}$ – $Dummy_{N6-N7}$

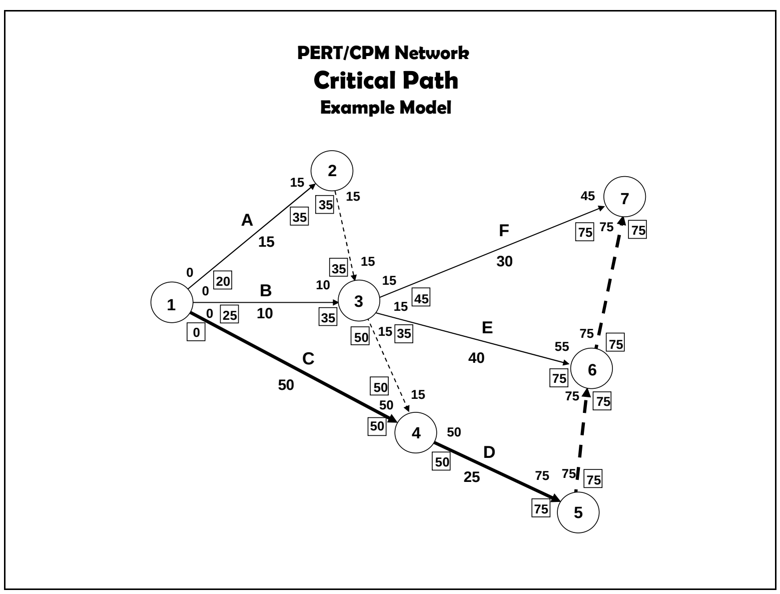

Figure 4
Example Model Critical Path

The Logic Method

Technique Algorithm

This approach relies on the same mathematical logic used in the **Forward Pass/Backward Pass** method. It is **seeking the path** that takes the **least amount of time** to reach the end of the network. This idea seems contrary to conventional thinking since **"least"** implies the **most efficient**. Here **"least"** means the **soonest** any path or the network can be completed or logically, it's **how long it will take** to complete a path or the project .

Applying this idea to each part and the entire network produces the following general rule that enables determination of the **Critical Path**:

> **Activities** that are on the **longest path between any two nodes** from the **Initial** to the **Terminal Nodes** are on the **Critical Path**.

This means that the rule must be applied to all paths between the **Initial** and **Terminal Nodes** to determine the **Critical Path**.

Longest Path Logic Application

The network shown in Figure 5 below will be used to demonstrate the application of this algorithm. This network appears busy, but using the **Logic Method** requires examining the **Network Internal Paths** which will help simplify its complexity.

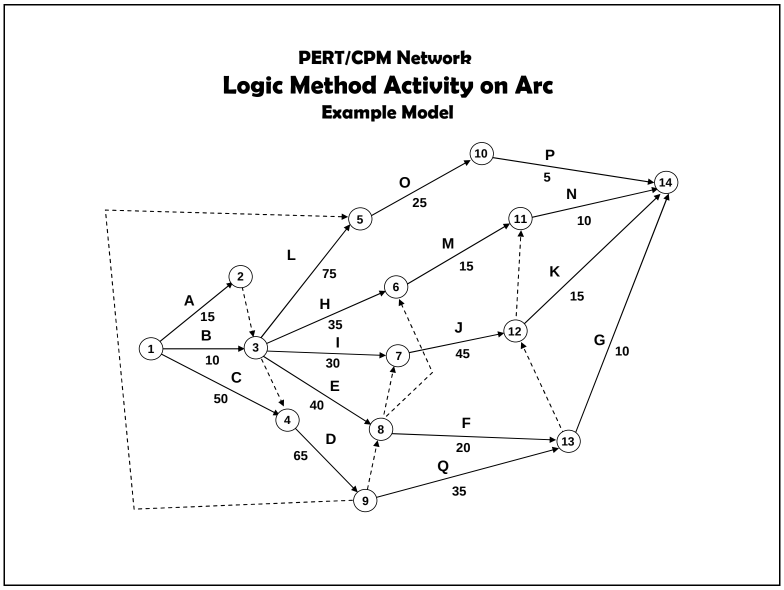

Figure 5
Example Model Critical Path

Since the **Initial Node** is **Node 1** and the **Terminal Node** is **Node 14**, the **Longest Path** between them will be the **Critical Path**. The **Critical Path duration** and **specific activities** are determined by comparing the **Longest Path** between any two nodes within the network.

This process can start anywhere in the network, but this example will begin at the **Initial Node** (Node 1). The first path duration comparison will between **Nodes 1** and **3** as shown in Figure 6 below.

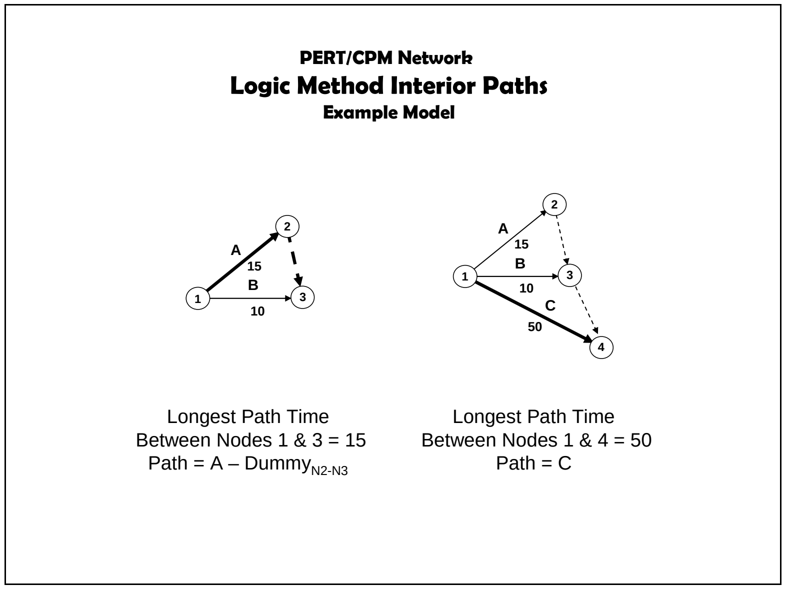

Figure 6
Example Model Interior Paths Node 1 to Nodes 3 and 4

The **two candidate paths** are **A – Dummy$_{N2-N3}$**, which is **15**, and **B**, which is **10**. Since **15** is the longest duration, the **Critical Path** from **Node 1** to **Node 3** is **A – Dummy$_{N2-N3}$**.

Looking again at Figure 6, the second path duration comparison is between **Nodes 1** and **4**. There are now **three candidate paths**: **A – Dummy$_{N2-N3}$ – Dummy$_{N3-N4}$**, which is **15**, **B – Dummy$_{N3-N4}$**, which is 10, and **C** which is **50**. Since **50** is the longest duration, the **Critical Path** to **Node 4** is **C**.

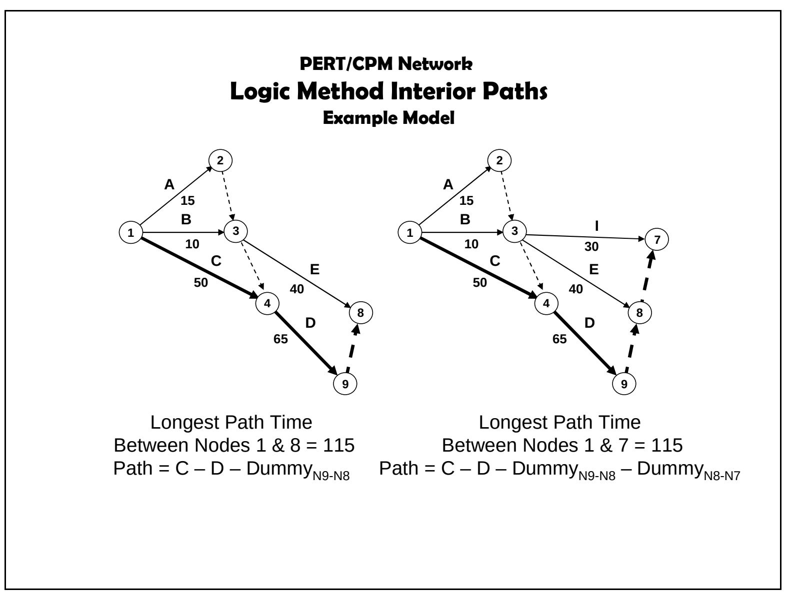

Figure 7
Example Model Interior Paths Node 1 to Nodes 8 and 7

Continuing the above process, the longest path across the entire network, as shown in Figures 6, 7, 8, 9, 10 and 11, with the candidate paths between nodes are summarized in Table 3 on page 11.

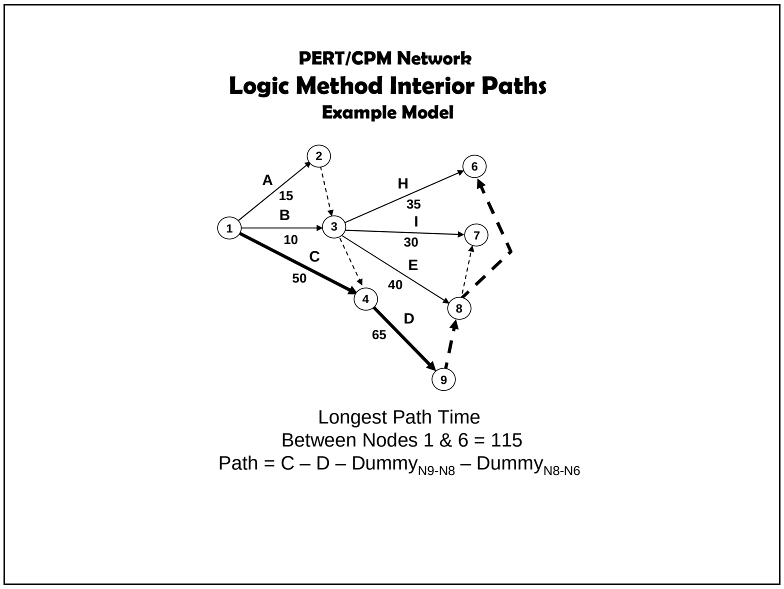

Figure 8
Example Model Interior Paths Node 1 to Node 6

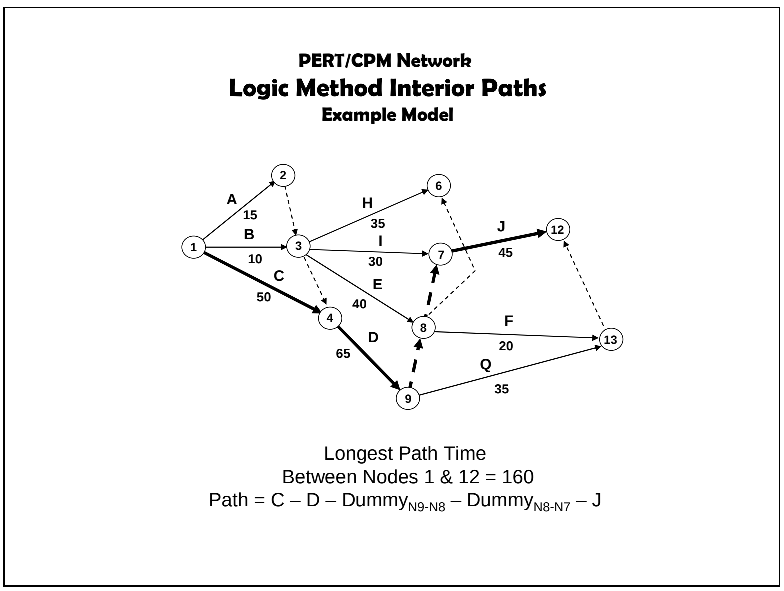

Figure 9
Example Model Interior Paths Node 1 to Node 12

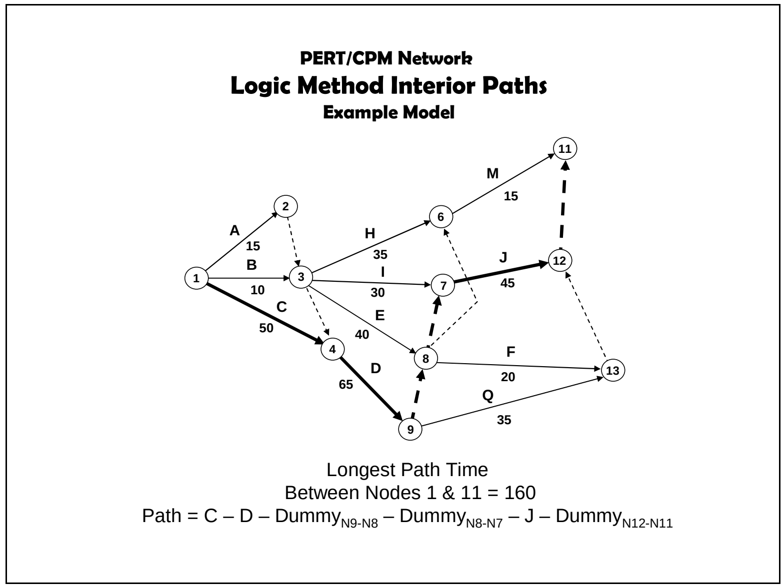

Figure 10
Example Model Interior Paths Node 1 to Node 11

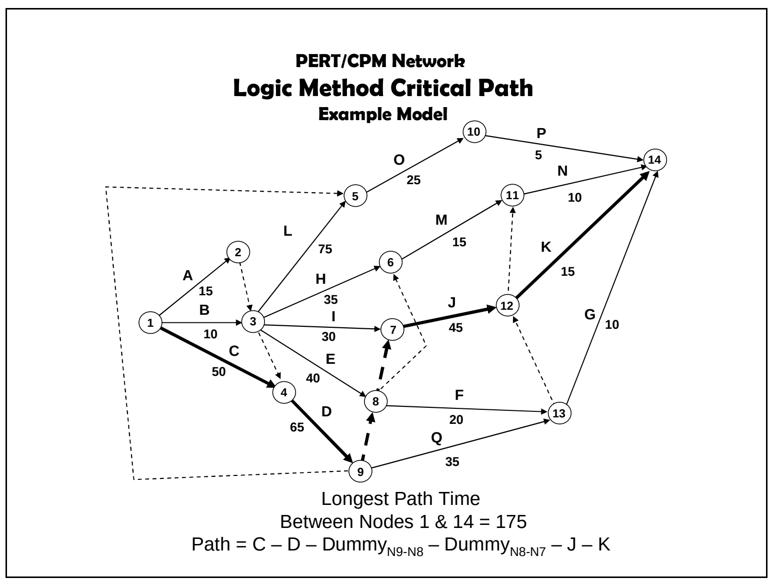

Figure 11
Example Model Critical Path Node 1 to Node 14

Path Outer Nodes		Candidate Paths	Candidate Path Duration	Longest Path
Start	Finish			
1	3	A - Dummy$_{N2-N3}$	15	15
		B	10	
1	4	A - Dummy$_{N2-N3}$- Dummy$_{N3-N4}$	15	50
		C	50	
1	8	A - Dummy$_{N2-N3}$- E	55	115
		C – D - Dummy$_{N9-N8}$	115	
1	7	A - Dummy$_{N2-N3}$ - I	45	115
		B – E - Dummy$_{N8-N7}$	50	
		C – D - Dummy$_{N9-N8}$ - Dummy$_{N8-N7}$	115	
1	6	A - Dummy$_{N2-N3}$- H	50	115
		B – E - Dummy$_{N8-N6}$	50	
		C – D - Dummy$_{N9-N8}$ - Dummy$_{N8-N6}$	115	
1	12	A - Dummy$_{N2-N3}$ – I - J	85	160
		B – E – F - Dummy$_{N12-N13}$	70	
		C – D – Q - Dummy$_{N9-N8}$	150	
		C – D - Dummy$_{N9-N8}$ – F - Dummy$_{N13-N12}$	135	
		C – D - Dummy$_{N9-N8}$ - Dummy$_{N8-N7}$ - J	160	
1	11	A - Dummy$_{N2-N3}$ – H - M	65	160
		A - Dummy$_{N2-N3}$ – I – J - Dummy$_{N12-N13}$	90	
		A - Dummy$_{N2-N3}$ – E – F - Dummy$_{N12-N13}$ - Dummy$_{N12-N11}$	75	
		C – D – Q - Dummy$_{N13-N12}$ - Dummy$_{N12-N11}$	150	
		C – D - Dummy$_{N9-N8}$ - Dummy$_{N8-N7}$ – J - Dummy$_{N12-N13}$	160	
1	14	A - Dummy$_{N2-N3}$ – L – O - P	120	175
		A - Dummy$_{N2-N3}$ – H – M - N	75	
		A - Dummy$_{N2-N3}$ – I – J - Dummy$_{N12-N13}$ - N	100	
		A - Dummy$_{N2-N3}$ – I – J - K	105	
		A - Dummy$_{N2-N3}$ – E – F – Dummy$_{N12-N13}$ - Dummy$_{N12-N11}$	85	
		A - Dummy$_{N2-N3}$ – E – F - G	85	
		A - Dummy$_{N2-N3}$ – I – J - K	105	
		A - Dummy$_{N2-N3}$ – E – F – Dummy$_{N12-N13}$ - K	90	
		C – D – Q - Dummy$_{N13-N12}$ - K	165	
		C – D – Q - G	160	
		C – D - Dummy$_{N9-N5}$ – O - P	145	
		C – D - Dummy$_{N9-N8}$ - Dummy$_{N8-N7}$ – J - K	175	
		C – D - Dummy$_{N9-N8}$ - Dummy$_{N8-N7}$ – J - Dummy$_{N11-N12}$ - N	170	

Table 3
Logic Method Internal and Critical Paths

The **Logic Method** as presented here may appear as cumbersome as **Forward Pass/Backward Pass**, but it is not. When applying this method, it is not necessary to write down the nodes and candidate paths shown in Table 3 because the **comparative process** can be done by inspection.

The determination of the **Critical Path** shown in Table 3 should only take a few minutes by inspection where **Forward Pass/Backward Pass** would take up to an hour to complete.

Determining Alternate Paths Slack

Slack is the amount of time **Non-critical Path Activities** have to **wait** until **Critical Path Activities** are **completed**.

This is where both methods came out about the same. A case can be made here that **Forward Pass/Backward Pass** is **easier** although its **difficulty occurs** when determining the **ES**, **EF**, **LF** and **LS** for each activity and the **Critical Path**.

Forward Pass/Backward Pass Method

The slack for any activity is determined with its **Start** and **Finish** values as follows:

Slack = LS-ES or LF-EF

This is demonstrated in the example problem shown in Figure 4 on page 7 and shown again in Figure 12 below.

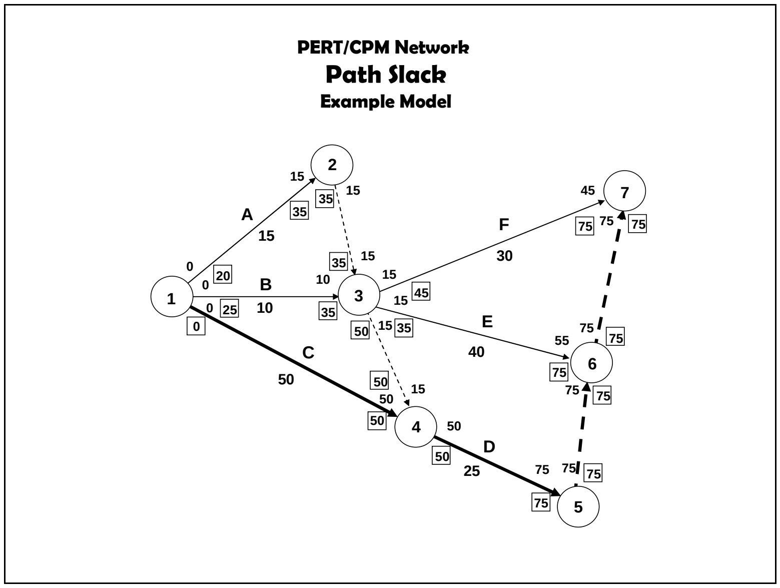

Figure 12
Example Model Forward Pass/Backward Pass Path Slack

Based on the above rule, the slack for all activities are shown in Table 4 below.

Activity	ES	LS	EF	LF	Slack
A	0	20	15	35	20
B	0	25	10	35	25
C	0	0	50	50	0
D	50	50	75	75	0
E	15	35	55	75	20
F	15	45	45	75	30

Table 4
Example Model Forward Pass/Backward Pass Activity Slack

Dummy activities **do not have any slack** since they have a **zero duration**.

Activities on the **Critical Path** also do not have any slack their **ES=LS** and **EF=LF**. This confirms the **Network Critical Path Determination** algorithm ("Each **Activity** with **Start values equal** and the **Finish Values equal** is on the **Critical Path.**") stated on page 6.

To determine the slack for any path, just add the slack for all activities on that path. This demonstrated in Table 5 below for selected paths in Figure 12.

Path	Activity Slack	Path Slack
A – Dummy$_{N2\text{-}N3}$ - F	20 + 30	50
B - F	25 + 30	55
B – E – Dummy$_{N6\text{-}N7}$	25 + 20	45

Table 5
Example Model Forward Pass/Backward Pass Path Selected Path Slack

Logic Method

This approach will not enable the determination of slack for individual activities. Only path slack can be ascertained. This is done by **subtracting the duration of any path** between **any two nodes** from the **longest path** between the **same two nodes**. This is shown in Table 6 below for selected paths in the Example Model shown in Figure 11 on page 11 and Figure 13 below.

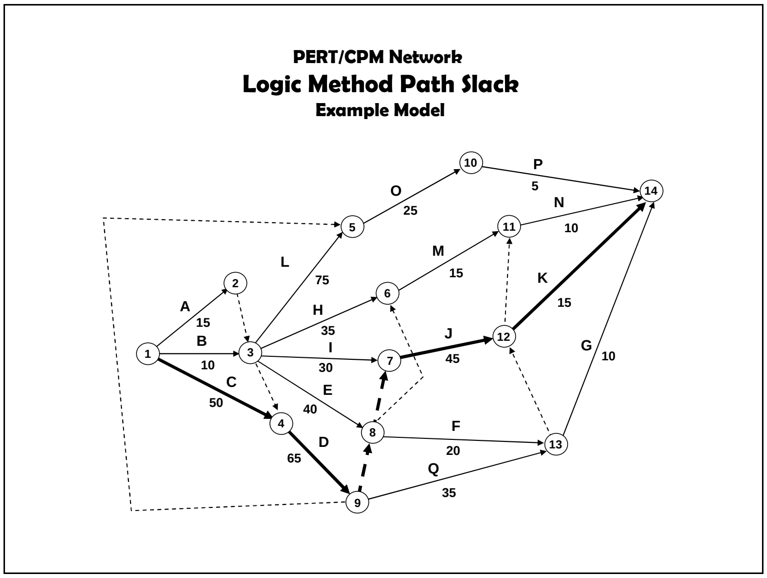

Figure 13
Example Model Logic Method Path Slack

Path Outer Nodes		Candidate Paths	Path Duration	Path Slack
Start	Finish			
3	14	L – O – P	105	20
		H – M – N	60	65
		I – J – K	90	35
		E – F – G	70	55
		Dummy$_{N3\text{-}N4}$ – D - Dummy$_{N9\text{-}N8}$ – F - G	105	20
		Dummy$_{N3\text{-}N4}$ – D - - Dummy$_{N9\text{-}N8}$ Dummy$_{N8\text{-}N6}$ – M - N	90	35
		E – Dummy$_{N8\text{-}N7}$ - J – Dummy$_{N12\text{-}N11}$ - N	95	80
		E – Dummy$_{N8\text{-}N7}$ - J – K	100	25
		E – Dummy$_{N8\text{-}N6}$ - M – N	65	110
		Dummy$_{N3\text{-}N4}$ – D - Q - G	110	15
		Dummy$_{N3\text{-}N4}$ – D - Dummy$_{N9\text{-}N8}$ – Dummy$_{N8\text{-}N7}$ - J - K	125	0

Table 6
Example Model Logic Method Selected Path Slack

Path Slack between nodes is functionally all that is needed since it will most likely be **distributed** among **all Activities** on the specific path anyway.

Advantages and Disadvantages of Both Techniques

Although each technique determines Critical Path when done properly and provides slack for all paths, they differ when applied as shown below.

Forward Pass/Backward Pass

Advantages

- Provides slack for individual activities
- Works with Microsoft Project Manager®

Disadvantages

- Takes a while (1-2 hours or more) to learn
- Prone to errors due to the tedium of the calculations
- Takes a while to apply to a large network (half hour or more)
- Cumbersome due to mathematical details
- Causes confusion when internal errors are made

Logic Method

Advantages

- Easy to learn (5 to 10 minutes)
- Quickly and easily determines Critical Path for any size network
- Clearly defines Critical Path
- Quickly determines Overall and Internal Paths Slack

Disadvantages

- Does not provide slack for individual activities
- Does not work with Microsoft Project Manager®

Index

A

Activity 1, 2, 3, 4, 5, 6, 7, 13
Activity Duration..2
Advantages..15
Advantages and Disadvantages of Both Techniques..15
Arc 1, 2

B

Backward Pass......................1, 2, 3, 5, 6, 12

C

completion node 4, 5
Critical Path 1, 5, 6, 7, 8, 9, 11, 12, 13, 15
Critical Path Method 1

D

deadline 3, 5
Disadvantages...15
Dummy Activities 2
Dummy Activity...............................2, 3, 4, 6
duration......................... 2, 3, 4, 5, 6, 8, 9, 14

E

Early Finish..................................3, 4, 5, 6, 7
Early Start..3, 4, 7

F

Forward Pass........ 1, 2, 3, 4, 5, 6, 12, 13, 15
Forward Pass/Backward Pass....2, 8, 12, 13, 15
Forward Pass/Backward Pass Algorithm 2, 3

I

Initial Node... 3, 8, 9

L

Late Finish 5, 6, 7
Late Start 5, 7
Latest Finish.. 3
Latest Start.. 3
Logic Method 1, 8, 12, 14, 15
Longest Path.. 8, 12
Longest Path Logic Application 8

M

Managerial Relevance 2
Microsoft Project Manager®..................... 15
most efficient.. 8

N

Network Critical Path Determination..... 7, 13
Network Initial Node................................... 4
Network Internal Paths 8
Network Symbols and Protocol................... 2
Network Terminal Node 5, 6
Nodes..................................... 2, 7, 9, 12, 14

P

Path Slack between nodes 14
PERT Network 1
PERT/CPM Network 2
Program Evaluation and Review Technique1
Project Earliest Finish 5
project management tools 1
project network model................................. 2

S

slack.. 13, 14, 15
soonest 3, 5, 8
Subsequent Activities 2

T

Technique Algorithm................................... 8
Terminal Node 5, 6, 8
The Logic Method 8, 12